AEDC-TR-87-36

C-3

User's Manual for EULER3DS

R. G. Hindman
Iowa State University
Ames, IA 50010

January 1988

Final Report for Period October 1986 – September 1987

Approved for public release; distribution unlimited.

ARNOLD ENGINEERING DEVELOPMENT CENTER
ARNOLD AIR FORCE BASE, TENNESSEE
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE

NOTICES

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Qualified users may obtain copies of this report from the Defense Technical Information Center.

References to named commercial products in this report are not to be considered in any sense as an endorsement of the product by the United States Air Force or the Government.

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

APPROVAL STATEMENT

This report has been reviewed and approved.

Keith L Kushman

KEITH L. KUSHMAN
Chief, Facility Technology Division
Directorate of Technology
Deputy for Operations

Approved for publication:

FOR THE COMMANDER

Marion L. Laster

MARION L. LASTER
Technical Director
Directorate of Technology
Deputy for Operations

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AEDC-TR-87-36	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION R.G. Hindman, Consultant 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Iowa State University Ames, IA 50010	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Arnold Engineering Development Center 8b. OFFICE SYMBOL (If applicable) DOF	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER CFSI84-5
8c. ADDRESS (City, State, and ZIP Code) Air Force Systems Command Arnold Air Force Base, TN 37389-5000	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 65807F; PROJECT NO. DB97VW; TASK NO.; WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) User's Manual for EULER3DS	
12. PERSONAL AUTHOR(S) Hindman, R.G., Consultant, Iowa State University	
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 10/86 TO 9/87
14. DATE OF REPORT (Year, Month, Day) January 1988	15. PAGE COUNT 34
16. SUPPLEMENTARY NOTATION Available in Defense Technical Information Center (DTIC).	

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP
20	04	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
three-dimensional space marching code
steady Euler equations
supersonic external flows

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Theoretical background is presented for a three-dimensional space marching code used to solve the steady Euler equations for supersonic external flows. The ALPHA scheme is introduced as a novel second-order integration scheme. The code is coupled to the QUICK-Geometry modeling system and incorporates either a perfect gas or equilibrium real gas model. A discussion of numerical aspects required for a user to set up the grid and the initial solution is presented.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL C.L. Garner	22b. TELEPHONE (Include Area Code) (615) 454-7813 22c. OFFICE SYMBOL DOCS

DD Form 1473, JUN 86 *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

PREFACE

The work reported herein was conducted by the author as principal investigator on subcontract CFSI84-5 between Iowa State University and Calspan Corporation/AEDC Division, operating contractor for the Aerospace Flight Dynamics testing effort at Arnold Engineering Development Center, Air Force Systems Command, Arnold Air Force Base, Tennessee, and on subcontracts CFSI85-7 and CFSI86-02 between The University of Texas at Arlington and Calspan Corporation/AEDC Division.

CONTENTS

Page

1.0 INTRODUCTION 5
2.0 GENERAL THEORY 5
2.1 The Governing Equations 5
2.2 The Space Marching Philosophy 6
2.3 Decoding 11
2.4 Shock Fitting 13
3.0 DISCRETIZATION 16
3.1 Geometric Related 16
3.1.1 Mesh Layout 16
3.1.2 Grid Generation 18
3.1.3 Body Surface Description 18
3.2 Integration Algorithm 18
3.2.1 The Generic Integration Scheme 18
3.2.2 Signal Splitting 20
3.2.3 The M-Flux 23
3.2.4 The U-Flux 24
3.2.5 The ALPHA-Scheme 25
3.3 Boundary Conditions 27
4.0 THE STARTUP PROBLEM 27
4.1 Initial Solution 27
5.0 INPUT DATA 28
5.1 Input Parameter Description 28
6.0 SUMMARY 29
7.0 REFERENCES 30

LIST OF FIGURES

1. General Six Sided Region 6
2. A General Quadrilateral Region on ζ^+ = constant 7
3. A Typical Non-Planar ζ^+ = constant Surface 9
4. Generic Finite Volume Approximation 10
5. Typical ξ = Constant Shock Surface 13
6. Mesh Layout 17
7. Discretized Generic Finite Volume Cell 19
8. Typical k^{th} ξ = Constant Celll Face 19
9. Cell Layout for the ALPHA-Scheme 25

A User's Manual For EULER3DS

by
Richard G. Hindman
Associate Professor of Aerospace Engineering
Iowa State University

1.0 INTRODUCTION

EULER3DS is a three dimensional space marching code for computing steady state solutions to the Euler equations applied to external flows. The code solves the integral equations of motion in a finite volume sense and is thus totally conservative. The algorithm used is called the ALPHA–scheme and was developed by the author for this code. This scheme is explicit with either first or second order accuracy and is constructed by splitting the fluxes on cell faces to account for upwind signal propagation. The code solves the governing equations using either a perfect gas model or an equilibrium air gas model due to Srinivasan (Ref. 1). There are six options regarding the grid generation schemes employed and the code is fully compatible with the QUICK geometry modeling system. The purpose of this document is to describe the code in a manner that will familiarize the user with the code's options and structure and enable relatively painless usage of the code.

2.0 GENERAL THEORY

2.1 The governing equations

The governing equations solved by **EULER3DS** are integral equations representing conservation of mass, momentum, and energy. These equations are now described. To begin with, we introduce the essential nomenclature. The dyadic $\underline{E}$ is comprised of the three Cartesian flux vectors and is given as

$$\underline{F} = \hat{i}e + \hat{j}f + \hat{k}g$$

where the vectors e,f and g are given by

$$e = \begin{bmatrix} \varrho u \\ p + \varrho u^2 \\ \varrho uv \\ \varrho uw \\ (\epsilon + p)u \end{bmatrix} \qquad f = \begin{bmatrix} \varrho v \\ \varrho uv \\ p + \varrho v^2 \\ \varrho vw \\ (\epsilon + p)v \end{bmatrix} \qquad g = \begin{bmatrix} \varrho w \\ \varrho uw \\ \varrho vw \\ p + \varrho w^2 \\ (\epsilon + p)w \end{bmatrix}$$

The unit vectors $\hat{i}, \hat{j}$, and $\hat{k}$ are the basis directions in the Cartesian x,y, and z coordinate system and u, v, and w are the corresponding velocity components. p is the static pressure and ρ is the fluid density. ϵ is the total energy per unit volume given by the equation

$$\epsilon = \varrho(e_i + .5(u^2 + v^2 + w^2))$$

where e_i is the specific internal energy. The governing integral equation is then given as

$$\int_V (\nabla \bullet \underline{F})\, dV = 0$$

This equation is transformed into a boundary integral equation using Green's theorem. The result is

$$\oint_{\partial V} \hat{n} \bullet \underline{F}\, dA = 0 \qquad (2.1)$$

In this equation, $\hat{n}$ is the outward unit normal vector to the boundary of the volume V. This boundary integral equation is discretized and solved on each of the finite volume cells. This discretization process is discussed in the next section.

It should be noted that in the actual solution algorithm, only four equations are solved since for the flows of interest, the total enthalpy is constant. However, this fact is not used in the development of the solution algorithm, thus making the algorithm more general.

2.2 The Space Marching Philosophy

For this discussion, consider the general six sided region shown in Figure 1.

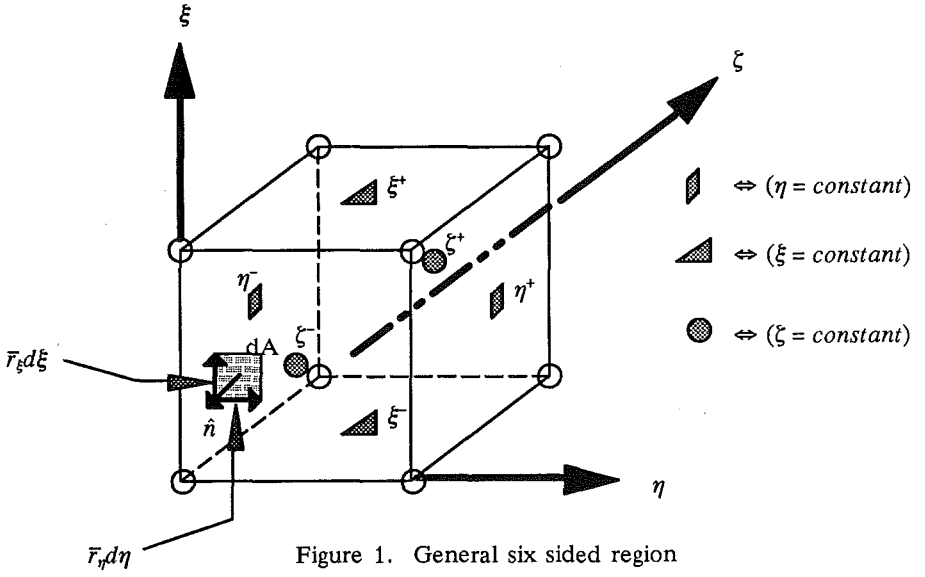


Figure 1. General six sided region

Equation (2.1) must be applied to a general region of this type. We first recognize that the quantity $\hat{n}dA$ is locally given by

$$\hat{n}dA = (\bar{r}_\xi \times \bar{r}_\eta)d\xi d\eta \quad on \quad \zeta^+$$

$$\hat{n}dA = (\bar{r}_\eta \times \bar{r}_\xi)d\xi d\eta \quad on \quad \zeta^-$$

$$\hat{n}dA = (\bar{r}_\eta \times \bar{r}_\zeta)d\eta d\zeta \quad on \quad \xi^+$$

$$\hat{n}dA = (\bar{r}_\zeta \times \bar{r}_\eta)d\eta d\zeta \quad on \quad \xi^-$$

$$\hat{n}dA = (\bar{r}_\zeta \times \bar{r}_\xi)d\zeta d\xi \quad on \quad \eta^+$$

$$\hat{n}dA = (\bar{r}_\xi \times \bar{r}_\zeta)d\zeta d\xi \quad on \quad \eta^-$$

where $\bar{r}$ is the local position vector. At this point, we choose ζ to be the marching direction. This means that ζ = constant surfaces are solution surfaces and we must solve for the ζ^+ solution surface in terms of everything else. This requires that the component of Mach number in the ζ direction be supersonic. We then restrict these ζ = constant surfaces to be planar and proceed to approximate the integrals over them. With reference to Figure 2, a second order evaluation on the ζ^+ surface is given by

$$\int_{\zeta^+} (\bar{r}_\xi \times \bar{r}_\eta) \cdot \underline{F}\, d\xi d\eta \simeq \frac{1}{2}\left[\bar{r}_\xi(\eta^-) \times \bar{r}_\eta(\xi^-) + \bar{r}_\xi(\eta^+) \times \bar{r}_\eta(\xi^+)\right] \cdot \underline{F_c}\, d\xi d\eta \qquad (2.2)$$

where $\underline{F_c}$ is the Cartesian flux dyadic at the quadrilateral centroid. The actual centroid location is

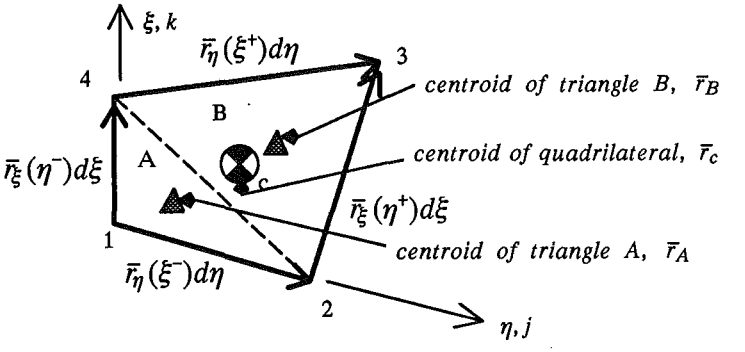


Figure 2. A general quadrilateral region on ζ^+ = *constant*

determined as follows. For the general planar area, S, the centroid is

$$\bar{r}_c = \frac{\int \bar{r} dS}{\int dS}$$

When this formula is applied to the quadrilateral in Fig. 2, we write

$$\bar{r}_c = \frac{\int \bar{r} dS}{\int dS} = \frac{\int \bar{r} dS_A + \int \bar{r} dS_B}{\int dS}$$

or

$$\bar{r}_c = \frac{S_A\left(\frac{\int \bar{r} dS_A}{S_A}\right) + S_B\left(\frac{\int \bar{r} dS_B}{S_B}\right)}{S}$$

or

$$\bar{r}_c = \left(\frac{S_A}{S}\right)\bar{r}_A + \left(\frac{S_B}{S}\right)\bar{r}_B$$

where S_A, S_B represent the areas of triangles A & B and $\bar{r}_A, \bar{r}_B$ represent the centroids of these triangles. Now the triangle centroids are given simply as

$$\bar{r}_A = \frac{1}{3}(\bar{r}_1 + \bar{r}_2 + \bar{r}_4)$$

and

$$\bar{r}_B = \frac{1}{3}(\bar{r}_2 + \bar{r}_3 + \bar{r}_4)$$

where $\bar{r}_{1,2,3,4}$ represent the position vectors to the corners of the quadrilateral region. The right hand side of Eq. (2.2) is simply expressed as

$$\bar{E} = \bar{\xi} \bullet \underline{F_c} = (\hat{\xi} \bullet \underline{F_c})A = \hat{E}A$$

where $\hat{\xi}$ is a unit vector perpendicular to ξ = constant surfaces and A is the area of the quadrilateral. The vector $\bar{\xi}$ is given by

$$\bar{\zeta} = \frac{1}{2}\{[\bar{r}_\xi(\eta^-) \times \bar{r}_\eta(\xi^-)] + [\bar{r}_\xi(\eta^+) \times \bar{r}_\eta(\xi^+)]\} d\xi d\eta$$

Next, consider a ξ = constant surface. Figure 3 illustrates this situation. This surface is, in general,

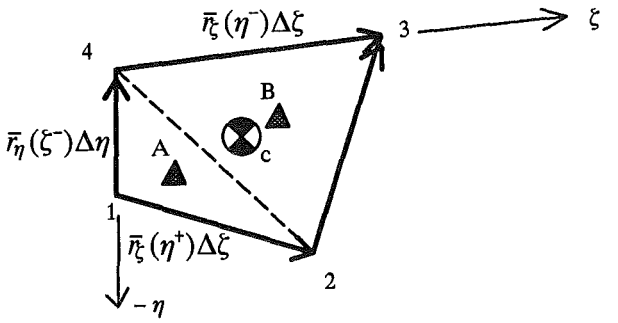


Figure 3. A typical non-planar $\xi = constant$ surface

non-planar. Consequently, its contribution to the integral equation (2.1) is slightly different than that for the ζ = constant surface. We seek to approximate

$$\int_{\xi^+} (\bar{r}_\eta \times \bar{r}_\zeta) \cdot \underline{F}\, d\eta\, d\zeta$$

on this non-planar quadrilateral. We define the quadrilateral centroid in just the same manner as previously done for the ζ = constant surface. Thus, the integral is approximated by

$$\widetilde{F} = \bar{\xi} \cdot \underline{F_c}$$

where

$$\bar{\xi} = \left\{\frac{1}{2}[\bar{r}_\eta(\zeta^-) \times \bar{r}_\zeta(\eta^+)] + \frac{1}{2}[(\bar{r}_\eta(\zeta^-) + \bar{r}_\zeta(\eta^+)) \times \bar{r}_\zeta(\eta^-)]\right\} d\eta d\zeta$$

Note that the bracketed terms in this expression represent the $\hat{n}dA$ for the triangles A & B of Figure 3. This equation is simplified to

$$\bar{\xi} = \frac{1}{2}\{[\bar{r}_\eta(\zeta^-) \times \bar{r}_\zeta(\eta^+)] + [\bar{r}_\eta(\zeta^-) \times \bar{r}_\zeta(\eta^-)] + [\bar{r}_\zeta(\eta^+) \times \bar{r}_\zeta(\eta^-)]\} d\eta d\zeta$$

Finally, for an η = constant surface, we define the appropriate integral

$$\int_{\eta^+} (\bar{r}_\zeta \times \bar{r}_\xi) \cdot \underline{F}\, d\zeta\, d\xi$$

to be

$$\tilde{G} = \bar{\eta} \cdot \underline{F_c}$$

where

$$\bar{\eta} = \frac{1}{2}\{[\bar{r}_\zeta(\xi^-) \times \bar{r}_\xi(\zeta^-)] + [\bar{r}_\zeta(\xi^-) \times \bar{r}_\xi(\zeta^+)] + [\bar{r}_\xi(\zeta^+) \times \bar{r}_\xi(\zeta^-)]\} d\zeta d\xi$$

The approximation to the integral equation, Eq. (2.1), is then given for the finite volume shown in Figure 4 as

$$\tilde{E}(\zeta^+) = \tilde{E}(\zeta^-) - [\tilde{F}(\xi^+) - \tilde{F}(\xi^-)] - [\tilde{G}(\eta^+) - \tilde{G}(\eta^-)] \tag{2.3}$$

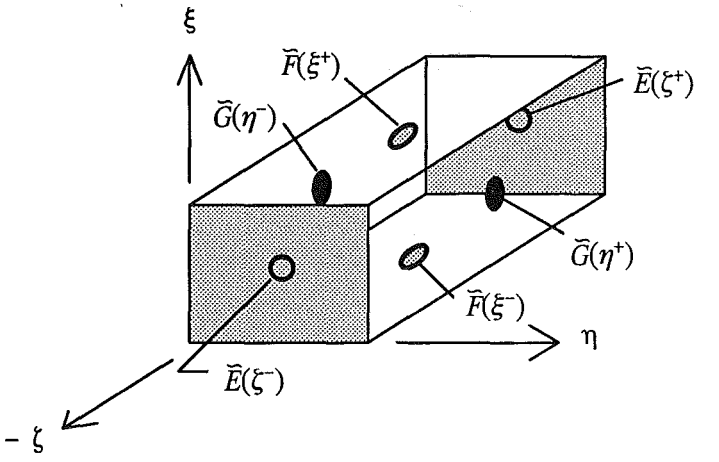


Figure 4. Generic finite volume approximation

from which it is clear that the value of $\tilde{E}$ at the next ζ = constant surface is computed at each integration step. This vector $\tilde{E}$ must then be decoded to yield the primitive variable information for the next step.

2.3 Decoding

The general decoding procedure applies to the vector $\bar{\zeta} \bullet \underline{F_c}$. We consider that we have taken an integration step and that $\bar{E}(\zeta^+)$ is known. We define the basis vector $\bar{\zeta}$ to be $\bar{\zeta} = (\zeta_1, \zeta_2, \zeta_3)$ and further define the contravariant velocity, W, to be

$$W = \bar{\zeta} \bullet \bar{q}$$

where $\bar{q}$ is the fluid velocity vector, $\bar{q}$ = (u,v,w). We then write

$$\bar{E}_1 = \varrho W$$

$$\bar{E}_2 = p\zeta_1 + \varrho u W = p\zeta_1 + u\bar{E}_1 \tag{2.4}$$

$$\bar{E}_3 = p\zeta_2 + \varrho v W = p\zeta_2 + v\bar{E}_1 \tag{2.5}$$

$$\bar{E}_4 = p\zeta_3 + \varrho w W = p\zeta_3 + w\bar{E}_1 \tag{2.6}$$

$$\bar{E}_5 = \frac{(p+\epsilon)}{\varrho}\bar{E}_1 = H_t\,\bar{E}_1 = \left(h + \frac{u^2+v^2+w^2}{2}\right)\bar{E}_1 \tag{2.7}$$

where h is the gas enthalpy and H_t is the constant total enthalpy. We define the ratio of enthalpy to internal energy to be

$$\bar{\gamma} = \frac{h}{e_i}$$

Thus,

$$h = \frac{\bar{\gamma} p}{(\bar{\gamma}-1)\varrho} = \frac{\bar{\gamma} p W}{(\bar{\gamma}-1)\bar{E}_1} \tag{2.8}$$

Equations (2.4) through (2.6) are solved for velocity components yielding

$$u = (\bar{E}_2 - p\,\zeta_1)/\bar{E}_1$$

$$v = (\bar{E}_3 - p\,\zeta_2)/\bar{E}_1 \tag{2.9}$$

$$w = (\bar{E}_4 - p\,\zeta_3)/\bar{E}_1$$

Substitution of these expressions into Eq. (2.8) yields

$$h = \frac{\bar{\gamma} p}{(\bar{\gamma}-1)\bar{E}_1^2}[\zeta_1\bar{E}_2 + \zeta_2\bar{E}_3 + \zeta_3\bar{E}_4 - p(\zeta_1^2+\zeta_2^2+\zeta_3^2)] \tag{2.10}$$

Substitution of Eqs. (2.9) and (2.10) into Eq. (2.7) yields a quadratic equation for pressure, p.

$$\left\{\frac{\bar{\gamma}+1}{2(\bar{\gamma}-1)}[\xi_1^2+\xi_2^2+\xi_3^2]\right\}p^2-\left\{\left(\frac{1}{\bar{\gamma}-1}\right)[\xi_1\bar{E}_2+\xi_2\bar{E}_3+\xi_3\bar{E}_4]\right\}p$$

$$+\left\{\bar{E}_1^2\ H_t-\frac{1}{2}(\bar{E}_2^2+\bar{E}_3^2+\bar{E}_4^2)\right\}=0$$

This closed form solution for pressure, P, is possible only in the case of $\bar{\gamma}$ = constant for a perfect gas. From this value of P, the density, ρ , is found from the equation

$$\varrho=\frac{\bar{E}_1^2}{\bar{E}_2\ \xi_1+\bar{E}_3\ \xi_2+\bar{E}_4\ \xi_3-p(\xi_1^2+\xi_2^2+\xi_3^2)} \tag{2.11}$$

which is obtianed from a linear combination of Eqs. (2.4), (2.5), and (2.6). The velocity components, u, v, and w are determined from Eq. (2.9).

For a real gas, an iterative procedure is employed. The first iteration uses $\bar{\gamma}$ to establish the initial guess. The function Q(p) is established to be

$$Q(p)=h+\left(\frac{u^2+v^2+w^2}{2}\right)-H_t \tag{2.12}$$

A Newton iteration yields

$$p^{m+1}=p^m-\frac{Q(p^m)}{Q'(p^m)}$$

where

$$Q'(p)=\frac{dQ}{dp}=\frac{dh}{dp}+u\frac{du}{dp}+v\frac{dv}{dp}+w\frac{dw}{dp} \tag{2.13}$$

The quantities $\frac{du}{dp}$, $\frac{dv}{dp}$, $\frac{dw}{dp}$ are obtained by differentiating Eqs. (2.9). $\frac{dh}{dp}$ is obtained by differentiating Eq. (2.10). This latter step involves the constant $\bar{\gamma}$. This influences the convergence path of the Newton iteration but not the result. The general iterative procedure then is to initialize p, ρ, u, v, w, using the constant $\bar{\gamma}$. h is determined by the real gas model as h = h(p,ρ). Equation

(2.12) gives a value of Q and Eq. (2.13) gives Q'. The update is performed generating a new value of p from which Eqs. (2.11) and (2.9) give ρ, u, v, w. This cycle continues until p ceases to vary.

2.4 Shock Fitting

EULER3DS is applicable to problems in which the Mach number in the ζ – direction is supersonic. Usually, external flow problems of this type have an outer shock boundary. As a result, the code includes a shock fitting algorithm which is now described.

The shock is assumed to be a ξ = constant surface which separates a uniform freestream flow from its "shocked" state. The ξ = constant surface is taken to lie just downstream of the shock so points on this surface are discontinuously related to the freestream through the Rankine–Hugoniot jump conditions. Consider the shock surface in Fig. 5.

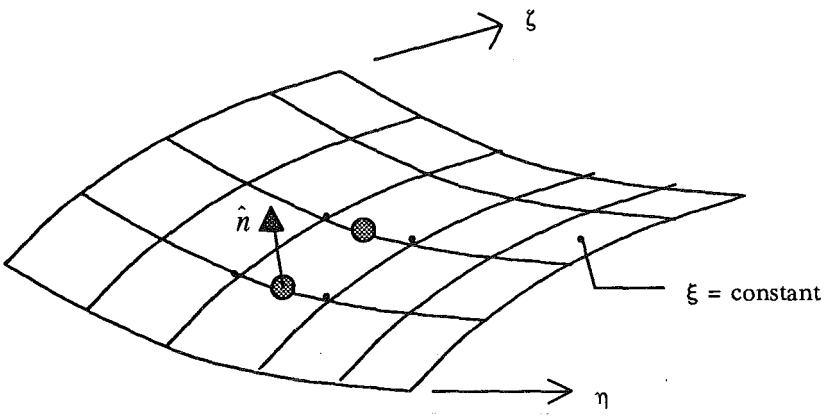


Figure 5. Typical ξ = constant shock surface

Assume conditions ahead of the shock (*eg.* : p_1, ϱ_1, h_1, $\overline{q}_1$) are known and that pressure, p_2 , behind the shock surface is known. The conservation equations for the fluid states across the shock are given as

$$\varrho_1 u_{1_N} = \varrho_2 u_{2_N}$$

$$u_{1_T} = u_{2_T}$$

$$p_1 + \varrho_1 u_{1_N}^2 = p_2 + \varrho_2 u_{2_N}^2$$

$$h_1 + \frac{u_{1_N}^2}{2} = h_2 + \frac{u_{2_N}^2}{2}$$

where subscripts N,T denote normal and tangential, respectively. Introducing $\overline{\gamma}_1$, $\overline{\gamma}_2$ to be

$$\bar{\gamma}_1 = \frac{h_1}{e_{i_1}} \quad , \quad \bar{\gamma}_2 = \frac{h_2}{e_{i_2}}$$

the following equation is obtained by combination of the above conservation laws.

$$u_{1_N}^2 = \frac{(\bar{\gamma}_1 - 1)(p_1 - p_2)}{2\varrho_1} \left\{ \frac{(\bar{\gamma}_2 + 1)p_2 + (\bar{\gamma}_2 - 1)p_1}{(\bar{\gamma}_2 - 1)p_1 - (\bar{\gamma}_1 - 1)p_2} \right\} \tag{2.14}$$

Combination of continuity and normal momentum also yields an equation for density, ϱ_2 .

$$\varrho_2 = \frac{(\varrho_1 u_{1_N})^2}{p_1 - p_2 + \varrho_1 u_{1_N}^2} \tag{2.15}$$

Thus, for a perfect gas $\bar{\gamma}_1 = \bar{\gamma}_2 = \gamma$ and Eq. (2.14) yields u_{1_N} when p_2 is known. For a real gas, the value of $\bar{\gamma}_2$ is unknown in Eq. (2.14). Consequently, an initial guess is made for $\bar{\gamma}_2$ and Eq. (2.14) yields a value of u_{1_N} . Equation (2.15) yields ϱ_2 . The combination of p_2 , ϱ_2 yields h_2 , e_{i_2} from which a new $\bar{\gamma}_2$ is determined. A repeat cycle yields a new value of ϱ_2 , etc. Ths is repeated until ϱ_2 converges.

Once this portion is completed, values of p_2, ϱ_2, $\bar{\gamma}_2$, h_2, e_{i_2}, u_{1_N} are known. The value of u_{1_N} is now used to determine how to propagate the shock point. In general, for the shock shown in Fig. 5,

$$\hat{n} = \frac{\bar{r}_\eta \times \bar{r}_\zeta}{|\bar{r}_\eta \times \bar{r}_\zeta|}$$

and

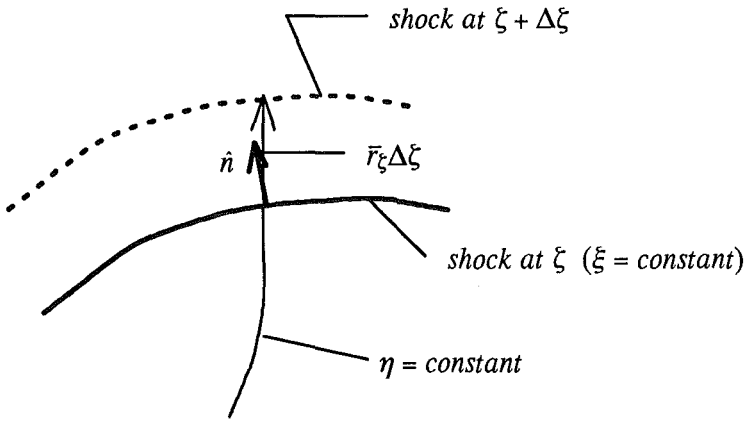
$$u_{1_N} = \bar{q}_1 \bullet \hat{n} \tag{2.16}$$

Since generally at any marching station, $\bar{r}_\eta$ is known and $\bar{q}_1$ is known, the vector $\bar{r}_\zeta$ is the quantity of concern in this equation. Equation (2.16) represents a single scalar equation with the three unknowns x_ζ, y_ζ, z_ζ . However, z_ζ controls the way in which the physical axial coordinate, z, is related to the computational marching coordinate, ζ , and as such, is specified. z_ζ = 1 in **EULER3DS** and consequently x_ζ, y_ζ are the two unknowns. An additional relationship is needed in order to solve for these quantities. This additional relationship controls the manner in which a shock point propagates relative to the existing shock surface. We have freedom in choosing exactly how to propagate a shock point. There are three options coded in **EULER3DS**. They are:

Option 1: motion along existing η = constant lines
Option 2: along a projection of $\hat{n}$ onto a ζ = constant surface
Option 3: along a cylindrical ray

These options are now described.

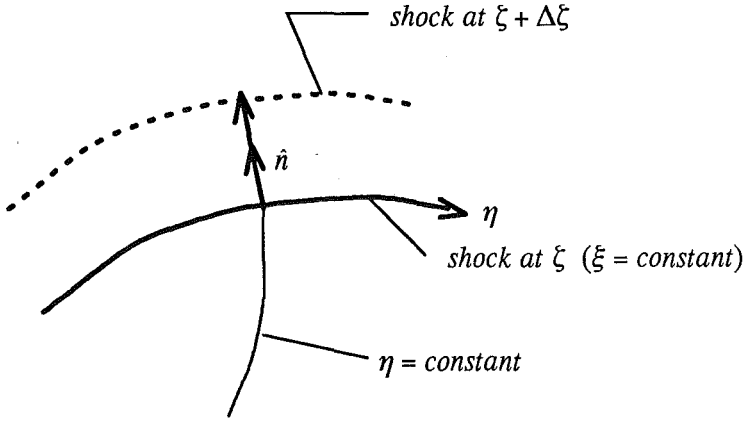
Option 1: In this case, that part of $\bar{r}_\zeta$ excluding the z_ζ part, say $\tilde{r}_\zeta$, must be aligned with $\bar{r}_\xi$.



Consequently, we may write $\tilde{r}_\zeta = s\bar{r}_\xi = (x_\zeta,\ y_\zeta)$. Substitution of this into Eq. (2.16) yields one equation with the single unknown, s. Solution for s then yields the shock slope vector, $\bar{r}_\zeta$. The shock position may be updated by integrating this vector.

Option 2: In this case $\tilde{r}_\zeta$ is expressed as

$$\tilde{r}_\zeta = s(\bar{r}_\eta \times \hat{k})$$



Again, s is the only unknown from which the shock slope $\bar{r}_\zeta$ is determined.

Option 3: In this case, $\bar{r}_\xi$ is given by $\bar{r}_\xi = s\hat{i}_r = s(\cos\phi\ \hat{i} + \sin\phi\ \hat{j})$ where ϕ is the meridional angle of the shock point in a cylindrical system defined by $x = r\cos\phi$, $y = r\sin\phi$. Again, s is the only unknown from which the shock slope, $\bar{r}_\xi$, is determined.

The shock fitting algorithm thus assumes a value of p_2 behind the shock. The remainder of the flow variables and the vector $\bar{r}_\xi$ are determined in this routine.

It should be noted that in the event that $p_2 < p_1$ at any given point during the calculation, the code sets $p_2 = p_1$ and constructs $\bar{r}_\xi$ in the normal way. The effect of this is that if a shock degenerates to a Mach surface, the solution is maintained correctly and the point is propagated along the Mach surface correctly.

3.0 DISCRETIZATION

The equatons and ideas discussed in Secton 2.0 are implemented on a discretized domain. The purpose of this section is to quantify this domain and quantify the numerical implementaion of the governing equations on this domain.

3.1 Geometric Related

First of all, the code assumes pitch plane symmetry. As mentioned, the solution to the governing flow equations is obtained by marching in the ζ (i.e., z) direction. Thus, a discretized grid or mesh is required at each ζ marching station. This mesh has as its four boundaries the following

$\xi_{min} \Leftrightarrow$ vehicle surface

$\xi_{max} \Leftrightarrow$ shock surface

$\eta_{min} \Leftrightarrow$ wind symmetry boundary

$\eta_{max} \Leftrightarrow$ lee symmetry boundary

The actual mesh layout is now described.

3.1.1 Mesh Layout

The 'corner grid' is defined to be that net of points located at the corners of the finite volume cells. The 'cell center grid' is that net of points occupying the cell centroid locations. The coordinates of both of these grids are known. However, the flow solution is generally known only on the 'cell center grid' except at boundaries.

Since the integration is performed in the ζ direction, a two-dimensional grid is required at each ζ-station. Figure 6 illustrates the typical grid layout for this code.

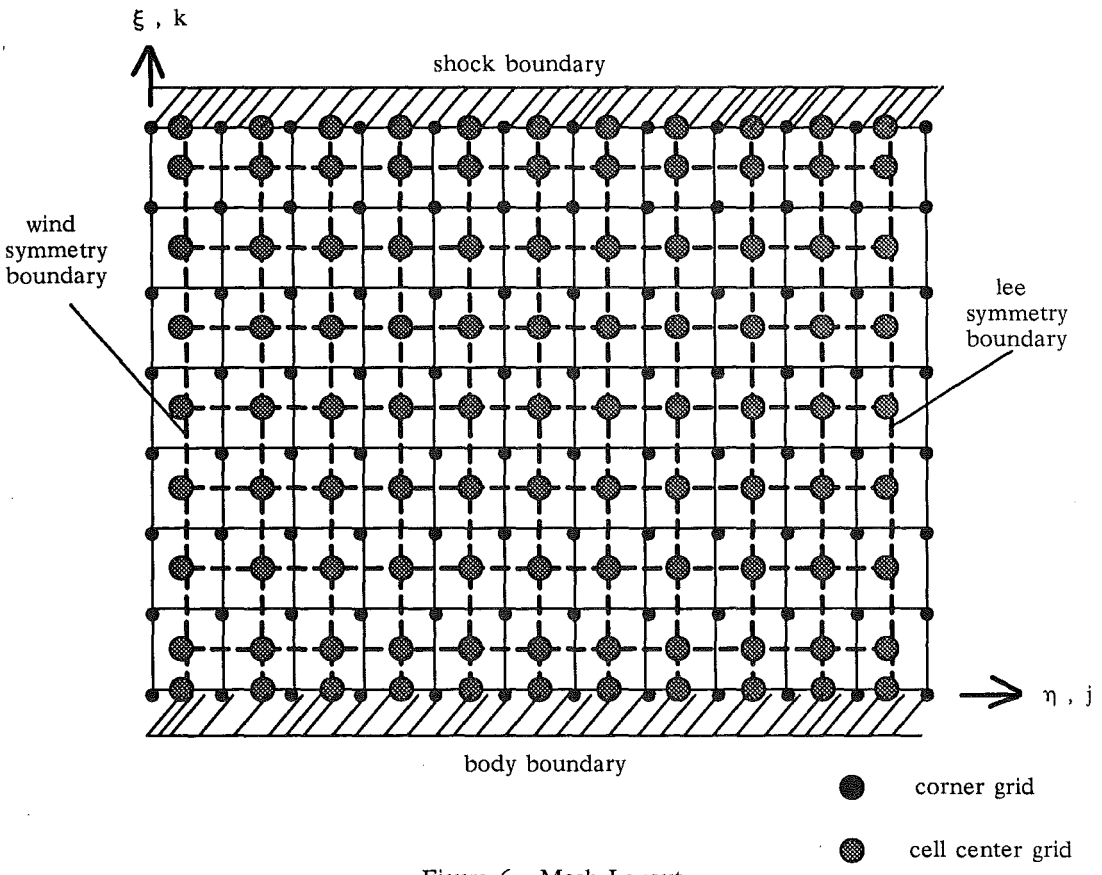


Figure 6. Mesh Layout

Notice that the centroid grid points exist on the symmetry boundaries, but the corner grid is reflected across these boundaries. Notice also that cell center grid points are defined on the body and the shock. This is to provide flow data on these surfaces instead of one half cell away. These points are actually not cell centers but rather segment centers. The coordinates of these points are simply the average of their two neighbors. Figure 6 indicates that the index j is associated with the η coordinate and k is asssociated with the ξ coordinate. The index n is used for the ζ coordinate. The 'corner grid' consists of JX by KX points including the reflected symmetry boundary points. This means that the 'centroid grid' contains JX1 by KXP1 points where JX1 = JX −1 and KXP1 = KX + 1. There are JX1 by KX1 finite volume cells (KX1 = KX − 1). The variables NTOT, NCATOT, and NCTOT are used in the code. They are defined as

NTOT = JX * KX
NCATOT = JX1 * KX1
NCTOT = JX1 * KXP1

Thus, there are NTOT 'corner grid' points, NCATOT cells, and NCTOT 'centroid grid' points. The flow solution is computed and stored at the NCTOT 'centroid grid' points. The geometric information is stored as follows :

XN, YN	NTOT 'corner grid' points at marching station, n.
X , Y	NTOT 'corner grid' points at marching station, n+1.
XCN, YCN	NCTOT 'centroid grid' points at n.
XC , YC	NCTOT 'centroid grid' points at n+1.
VCELLN	NCATOT cell areas at n.
VCELL	NCATOT cell areas at n+1.

3.1.2 Grid Generation

The corner grid is determined by a general grid generation algorithm with several options. These options include the scheme of Winslow (Ref. 2), Middlecoff's (Ref. 3) application of Thompson's (Ref. 4) scheme, Noack's algebraic scheme (Ref. 5), a simple algebraic scheme, and Sorenson's (Ref. 6) application of Thompson's scheme. All of these grid generation options require a specification of both body surface points and shock points. The body surface points are determined by the QUICK geometry modeling system (Refs. 7–9) and the shock points are determined by integration of shock slopes determined in the shock fitting scheme discussed in section 2.4. Once the corner grid is known, the cell center grid is determined by constructing the coordinates of all of the quadrilateral centroids as described in section 2.2.

3.1.3 Body Surface Description

EULER3DS is designed with only one subroutine which interfaces the code to the particular geometry being solved. BPOINT takes as input the value of z (or ζ) and returns JX values of x,y which represent the body cross section coordinates at this z station. The meridional placement of these grid points on a given body cross section is based on equal arc length in the η direction. BPOINT is interfaced to the QUICK geometry modeling system. For the given z value, BPOINT breaks the cross section contour into NPMAX points (NPMAX presently equals 1000) spaced at equal meridional angle increments. The coordinates of these sample points in this fine database are obtained by calling the QUICK subroutine CSGEOM. BPOINT then interpolates the JX grid points spaced according to equal arc length onto this fine database thus providing the body surface 'corner grid' points.

3.2 Integration Algorithm

This section expands on the details of the novel integration algorithm used in **EULER3DS.** We begin with the generic integration scheme applied to Equation (2.3) (repeated here for convenience).

$$\bar{E}(\zeta^+) = \bar{E}(\zeta^-) - [\bar{F}(\xi^+) - \bar{F}(\xi^-)] - [\bar{G}(\eta^+) - \bar{G}(\eta^-)] \tag{3.1}$$

3.2.1 The Generic Integration Scheme

Consider Figure 7 as a reference. Equation (3.1) is written in discretized form as

$$\tilde{E}^{n+1}_{j,k+1} = \tilde{E}^{n}_{j,k+1} - \left(\tilde{F}^{n+1/2}_{j+1/2,k+1} - \tilde{F}^{n+1/2}_{j+1/2,k}\right) - \left(\tilde{G}^{n+1/2}_{j+1,k+1/2} - \tilde{G}^{n+1/2}_{j,k+1/2}\right) \tag{3.2}$$

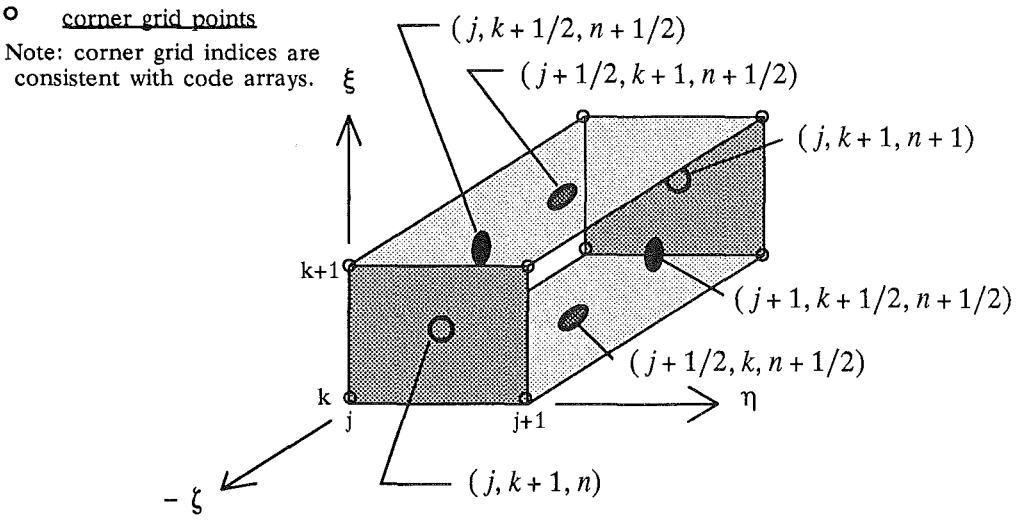


Figure 7. Discretized generic finite volume cell

where $1 \le j \le jx1$ and $1 \le k \le kx1$. This equation thus produces a solution at all cell center points. The body and shock points are treated separately. This scheme obviously cannot be implemented without some decision about how to evaluate the $\tilde{F}$, $\tilde{G}$ quantities at the cell face centers. Consider first the k^{th} ξ = constant cell face on which we wish to evaluate $\tilde{F}^{n+1/2}_{j+1/2,k}$ (see Fig. 8).

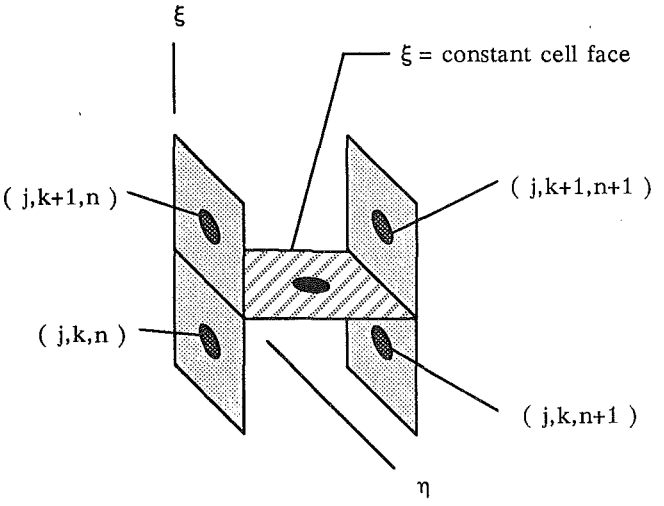


Figure 8. Typical k^{th} ξ = constant cell face

From this sketch, it is clear that the four closest centroid points at which the flow data are known are those points at (j,k,n), (j,k,n+1), (j,k+1,n), and (j,k+1,n+1). The value of $\tilde{F}^{n+1/2}_{j+1/2,k}$ is thus obtained from data at these four points. Since the data at the two (n+1) points is not really known apriori, estimates which come from a predictor step are used. This makes the overall scheme a

predictor–corrector scheme. The rationale for choosing which of the four points to use in computing $\tilde{F}^{n+1/2}_{j+1/2,k}$ is based upon the signal propagation mechanisms within the fluid. This subject is now discussed.

3.2.2 Signal Splitting

The vector $\bar{F}$ is a linear combination of e, f, and g and is written as $\bar{\xi} \cdot \underline{F_c}$ or

$$\tilde{F} = \xi_1 e + \xi_2 f + \xi_3 g$$

The vectors e, f, and g may be written as

$$e = AU \quad , \quad f = BU \quad , \quad g = CU$$

where U = (ρ , ρu , ρv , ρw , ϵ) and A, B, and C are rotation matrices which coincide with the Jacobian matrices $\frac{de}{dU}$, $\frac{df}{dU}$, and $\frac{dg}{dU}$, respectively, for the case of a perfect gas. For a real gas, $A \neq \frac{de}{dU}$, etc., but e = AU still holds for the appropriate choice of A. This choice is simply the Jacobian A for a perfect gas with γ replaced by $\frac{h}{e_i}$. Thus, if we define a matrix B^* as

$$B^* = \xi_1 A + \xi_2 B + \xi_3 C$$

then we may write

$$\tilde{F} = B^* U \tag{3.3}$$

In a like manner, we may obtain

$$\tilde{E} = A^* U \tag{3.4}$$

where $A^* = \zeta_1 A + \zeta_2 B + \zeta_3 C$ and $\bar{\zeta}$ = $(\zeta_1 , \zeta_2 , \zeta_3)$. Combination of Eqs. (3.3) and (3.4) yields

$$\tilde{F} = B^* (A^*)^{-1} \tilde{E}$$

Since $\bar{E}$ is the dependent variable vector and the flux, $\bar{F}$, is expressed as a rotation of $\bar{E}$ through the diagonalizable matrix $B^* A^{*-1}$, $\bar{F}$ may be split into components according to the system characteristics (eigenvalues of $B^* A^{*-1}$). In fact, $B^* A^{*-1}$ may be written as

$$B^* A^{*-1} = T \Lambda T^{-1}$$

where Λ is a diagonal matrix containing the eigenvalues, λ_i , of $B^* A^{*-1}$, T is a matrix whose columns contain the right eigenvectors, r_i , of $B^* A^{*-1}$, and T^{-1} is a matrix whose rows contain the left eigenvectors, l_i , of $B^* A^{*-1}$.

The eigenvalues of $B^* A^{*-1}$ are given by

$$\lambda_{1,2,3} = \frac{\mathcal{V}}{\mathcal{U}}$$

$$\lambda_{4,5} = \frac{\mathcal{U}\,\mathcal{V} - c^2(\bar{\xi}\bullet\bar{\xi}) \pm c\sqrt{RAD}}{\mathcal{U}^2 - c^2(\bar{\xi}\bullet\bar{\xi})}$$

$$RAD = c^2(\bar{\xi}\bullet\bar{\xi})^2 - 2\mathcal{U}\,\mathcal{V}(\bar{\xi}\bullet\bar{\xi}) + \mathcal{U}^2(\bar{\xi}\bullet\bar{\xi}) + \mathcal{V}^2(\bar{\xi}\bullet\bar{\xi}) - c^2(\bar{\xi}\bullet\bar{\xi})(\bar{\xi}\bullet\bar{\xi})$$

where $\mathcal{U} = (\bar{\xi}\bullet\vec{q})$, $\mathcal{V} = (\bar{\xi}\bullet\vec{q})$ and c is the sonic velocity. The left eigenvectors, l_i , are obtained by solving

$$(B^*(A^*)^{-1} - \lambda_i \mathrm{I})^T l_i = 0 \tag{3.5}$$

This is a formidable task. A simpler system to solve is obtained by recognizing that

$$A = M\hat{A}M^{-1}$$

where M represents a similarity transformation and $\hat{A}$ is a Jacobian matrix $\frac{de}{d\hat{U}}$ where $\hat{U}$ is the primitive variable vector. Likewise,

$$B = M\hat{B}M^{-1} \quad \text{and} \quad C = M\hat{C}M^{-1}$$

from which $A^* = M\hat{A}^*M^{-1}$ $B^* = M\hat{B}^*M^{-1}$

where $\hat{A}^* = \zeta_1\hat{A} + \zeta_2\hat{B} + \zeta_3\hat{C}$ and $\hat{B}^* = \xi_1\hat{A} + \xi_2\hat{B} + \xi_3\hat{C}$.

Thus we write

$$B^*A^{*-1} = M(\hat{B}^*\hat{A}^{*-1})M^{-1}$$

from which Eq. (3.5) becomes

$$(\hat{B}^* - \lambda_i \hat{A}^*)^T \tilde{l}_i = 0$$

where $\tilde{l}_i = M^T l_i$ .The vectors $\tilde{l}_i$ are more easily determined and are found to be

$$\tilde{l}_1^T = (a_{11}^2c^2\ ,\ 0\ ,\ a_{13}c^2\ ,\ -a_{12}c^2\ ,\ -a_{11}^2)$$
$$\tilde{l}_2^T = (-a_{11}a_{12}c^2\ ,\ a_{13}c^2\ ,\ 0\ ,\ -a_{11}c^2\ ,\ a_{11}a_{12})$$
$$\tilde{l}_2^T = (a_{11}a_{13}c^2\ ,\ a_{12}c^2\ ,\ -a_{11}c^2\ ,\ 0\ ,\ -a_{11}a_{13})$$
$$\tilde{l}_4^T = (0\ ,\ -\varrho c^2a_{41}\ ,\ -\varrho c^2a_{42}\ ,\ -\varrho c^2a_{43}\ ,\ \mathcal{V} - \lambda_4\mathcal{U}\ )$$
$$\tilde{l}_5^T = (0\ ,\ -\varrho c^2a_{51}\ ,\ -\varrho c^2a_{52}\ ,\ -\varrho c^2a_{53}\ ,\ \mathcal{V} - \lambda_5\mathcal{U}\ )$$

The matrix M is well known to be

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \varrho & 0 & 0 & 0 \\ v & 0 & \varrho & 0 & 0 \\ w & 0 & 0 & \varrho & 0 \\ \dfrac{u^2+v^2+w^2}{2} & \varrho u & \varrho v & \varrho w & \dfrac{1}{\gamma-1} \end{bmatrix}$$

From this l_i may be determined. The a_{jk} in the $\bar{l}_i$ vectors is defined to be $a_{jk} = \xi_k - \lambda_j \zeta_k$.

The right eigenvectors, r_i , are obtained by solving

$$(B^*(A^*)^{-1} - \lambda_i \mathrm{I})r_i = 0$$

Again a simpler system is obtained as before resulting in

$$(\hat{B}^* - \lambda_i \hat{A}^*)\bar{\bar{r}}_i = 0$$

from which $r_i = M \hat{A}^* \bar{\bar{r}}_i$. The $\bar{\bar{r}}_i$ are found to be

$$\bar{\bar{r}}_1{}^T = \frac{1}{D}(1\ ,\ 0\ ,\ a_{13}\ ,\ -a_{12}\ ,\ 0)$$

$$\bar{\bar{r}}_2{}^T = \frac{1}{a_{11}D}(-a_{12}\ ,\ a_{11}a_{13}\ ,\ 0\ ,\ -a_{11}^2\ ,\ 0)$$

$$\bar{\bar{r}}_3{}^T = \frac{1}{a_{11}D}(a_{13}\ ,\ a_{11}a_{12}\ ,\ -a_{11}^2\ ,\ 0\ ,\ 0)$$

$$\bar{\bar{r}}_4{}^T = \frac{1}{D_4^*}\left(\frac{\varrho\,(a_{41}^2 + a_{42}^2 + a_{43}^2)}{\overline{\nabla} - \lambda_4 \overline{\mathrm{U}}}\ ,\ -a_{41}\ ,\ -a_{42}\ ,\ -a_{43}\ ,\ \varrho\,(\overline{\nabla} - \lambda_4 \overline{\mathrm{U}})\right)$$

$$\bar{\bar{r}}_5{}^T = \frac{1}{D_5^*}\left(\frac{\varrho\,(a_{51}^2 + a_{52}^2 + a_{53}^2)}{\overline{\nabla} - \lambda_5 \overline{\mathrm{U}}}\ ,\ -a_{51}\ ,\ -a_{52}\ ,\ -a_{53}\ ,\ \varrho\,(\overline{\nabla} - \lambda_5 \overline{\mathrm{U}})\right)$$

where $D = \overline{\mathrm{U}}\, c^2(a_{11}^2 + a_{12}^2 + a_{13}^2)$

and $D_{4,5}^* = 2\varrho\, c^2[\overline{\mathrm{U}}\,(\bar{\xi} \bullet \bar{\xi} - \lambda_{4,5}\bar{\xi} \bullet \bar{\zeta}) - \overline{\nabla}\,(\bar{\xi} \bullet \bar{\zeta} - \lambda_{4,5}\bar{\zeta} \bullet \bar{\zeta})]$.

From this, the r_i are determined.

The diagonalization of the matrix $B^* A^{*-1}$ allows us to write

$$\tilde{F} = \tilde{F}_1 + \tilde{F}_2 + \tilde{F}_3 + \tilde{F}_4 + \tilde{F}_5 = \sum_{i=1}^{5} \tilde{F}_i$$

where $\tilde{F}_i = T\Lambda_i T^{-1}\tilde{E}$ and Λ_i is zero everywhere except in position (i,i) where the element λ_i exists. With some manipulation, it is easily shown that this simplifies to

$$\tilde{F}_i = \lambda_i \, r_i \, l_i^T \, \tilde{E} \tag{3.6}$$

which is interesting since it shows that the flux vector component, $\tilde{F}_i$, has the direction of the i^{th} right eigenvector. Since $\lambda_i \, l_i^T = l_i^T \, B^* A^{*-1}$, Eq. (3.6) is also written as

$$\tilde{F}_i = r_i \, l_i^T \, \tilde{F}$$

The significance of writing $\tilde{F} = \sum_{i=1}^{5} \lambda_i \, r_i \, l_i^T \, \tilde{E}$ is that each characteristic piece of $\tilde{F}$ is identified and the evaluation of these pieces can take into account the signal propagation directions for these flux contributions.

We began with the problem of determining $\tilde{F}$ on the k^{th} ξ = constant cell face. We now have the problem of determining five values of $\tilde{F}_i$ on this face. Two candidate evaluations of the $\tilde{F}_i$ are introduced in the following two sections.

3.2.3 The M-Flux

The M-flux is styled after a finite volume version of the MacCormack scheme as applied to each of the $\tilde{F}_i$. We will denote this flux as $\tilde{F}_M = \sum_{i=1}^{5} \tilde{F}_{M_i}$. With reference to Figure 8, we define $\hat{\lambda}_i$ to be λ_i calculated using the $\bar{\xi}$, $\bar{\zeta}$ values corresponding to the cell face under consideration and the primitive variables on that face which are obtained by averaging the nearby neighbors. Thus

$$\hat{\lambda}_i = \left[\lambda_i \left(\frac{1}{2}\left(\hat{U}_{j,k} + \hat{U}_{j,k+1} \right) , \; \bar{\xi} , \; \bar{\zeta} \right) \right]$$

where $\hat{U}$ is the primitive variable vector. Also,

$$\sigma_i = \frac{1}{2}\left(1 - sgn(\hat{\lambda}_i) \right)$$

Note that $\hat{\lambda}_i$ is an effective eigenvalue on the ξ = constant cell face. Then we write

$$\tilde{F}_{M_i} = \frac{sgn|\hat{\lambda}_i|}{2} \left[(\tilde{F}_i)^n_{j,k+\sigma_i} + (\tilde{F}_i)^{\overline{n+1}}_{j,k+1-\sigma_i} \right] \tag{3.7}$$

Thus, if the i^{th} eigenvalue is positive ($\hat{\lambda}_i > 0$), the information for $\tilde{F}_i$ on the k^{th} ξ = constant cell face comes from points "upwind" at n and "downwind" at n+1 or along the characteristic. The $sgn|\hat{\lambda}_i|$ coefficient is necessary to cause $\tilde{F}_{M_i} = 0$ in the event that $\hat{\lambda}_i = 0$ as indicated by Eq. (3.6). The quantity evaluated at $\overline{n+1}$ results from a predictor step in which

$$\tilde{F}_i = sgn|\hat{\lambda}_i|(\tilde{F}_i)^n_{j,k+\sigma_i} \tag{3.8}$$

In practice, the evaluation of Eq. (3.7) is not unique since $\tilde{F}_i = \lambda_i\, r_i\, l_i^T\, \tilde{E}$. Several methods of evaluation were tried and one clearly worked best. The portions of Eq. (3.6) which involve the geometric data, $\bar{\xi}$ and $\bar{\eta}$, are unique and therefore result in no ambiquity. However, the parts which involve the flow variables may be constructed in a number of ways. If we seek evaluation of $\tilde{F}_i$ on the ξ = constant face under investigation then $\bar{\xi}$, $\bar{\zeta}$ are clearly known at level n and are known for level $\overline{n+1}$ once the $\overline{n+1}$ grid is established. If we discover that $\hat{\lambda}_i$ for this face is positive, for example, then we evaluate $\tilde{F}_i$ as

$$\tilde{F}_i = \lambda_i\, r_i\, l_i^T\, \tilde{E}$$

where $\lambda_i = \hat{\lambda}_i$ and $r_i\, l_i^T\, \tilde{E}$ is evaluated using $\bar{\xi}$, $\bar{\zeta}$ at n or $\overline{n+1}$ as appropriate, and the primitive variables from point (j,k) for the level n value, and point (j,k+1) for the level $\overline{n+1}$ value.

It must be noted that the M–flux evaluation of the $\tilde{F}_i$ is perfectly consistent with a characteristic like method even though the $\overline{n+1}$ quantity is 'downwind'.

3.2.4 The U–Flux

The U–flux evaluation candidate results from a belief that all information must come from 'upwind' defined in the following sense: If $\lambda^n > 0$ at a point then all information for the integration step to n+1 must come from the negative coordinate increment side of this point. This attitude is ridiculous unless λ = 0 but never–the–less, the result of this thinking is an evaluation of $\tilde{F}_{U_i}$ as

$$\tilde{F}_{U_i} = \frac{sgn|\hat{\lambda}_i|}{2}\left[(\tilde{F}_{i_e})^n_{j,k+1-\sigma_i} + (\tilde{F}_i)^{\overline{n+1}}_{j,k+\sigma_i}\right] \tag{3.9}$$

where subscript e denotes 'extrapolated value' and

$$(\tilde{F}_{i_e})^n_{j,k+1-\sigma_i} = 2(\tilde{F}_i)^n_{j,k+\sigma_i} - (\tilde{F}_i)^n_{j,k+\sigma_i - sgn(\hat{\lambda}_i)} \tag{3.10}$$

As in the case of the M–flux, practice has shown that a better formula is to use the appropriate $\bar{\xi}$, $\bar{\zeta}$ vectors for the cell face under consideration and use the primitive variables from the 'upwind' point indicated but use $\hat{\lambda}_i$ instead of λ_i at the upwind point. So, for instance, $(\tilde{F}_i)^n_{j,k+\sigma_i}$ in Eq. (3.10) is evaluated as

$$(\tilde{F}_i)^n_{j,k+\sigma_i} = (\hat{\lambda}_i)(r_i\, l_i^T\, \tilde{E})_{j,k+\sigma_i}$$

where the left and right eigenvectors and $\tilde{E}$ use primitive variables from point (j,k+σ_i) and $\bar{\xi}$, $\bar{\zeta}$ values appropriate for the cell face in question. As in the case of the M–flux, the quantities required at $\overline{n+1}$ are assumed to be known via an upwind predictor step in which $\tilde{F}_i$ is obtained by Eq. (3.8). Either of the M–flux or the U–flux method will produce stable numerical results. These two schemes, however, exhibit opposite phase error characteristics. It is for this reason that these two schemes are

combined into a one parameter family of schemes in an attempt to produce a scheme superior to either of the individual schemes. The result is called the ALPHA–scheme which is the subject of the next section. Fromm (Ref. 10) tried something similar to this for a simple scalar linear equation.

3.2.5 *The ALPHA–scheme*

To begin the discussion of the ALPHA–scheme, we simply define the flux on the typical ξ = constant cell face to be a linear combination of the M–flux and the U–flux. That is, we take

$$\tilde{F}_i = \alpha\, \tilde{F}_{M_i} + (1 - \alpha)\, \tilde{F}_{U_i}$$

When α = 1, we use entirely the M–flux and α = 0 gives the U–flux. A value of α = .5 weights each part equally and exhibits superior dispersive properties to either M or U separately. Experience has shown that for either M or U separately, dispersion grows dramatically worse as the local Courant number becomes small. The α–scheme with α =.5, however, is quite insensitive to Courant number.

This section is ended with a summary of the overall integration algorithm. Figure 9 is used as a reference.

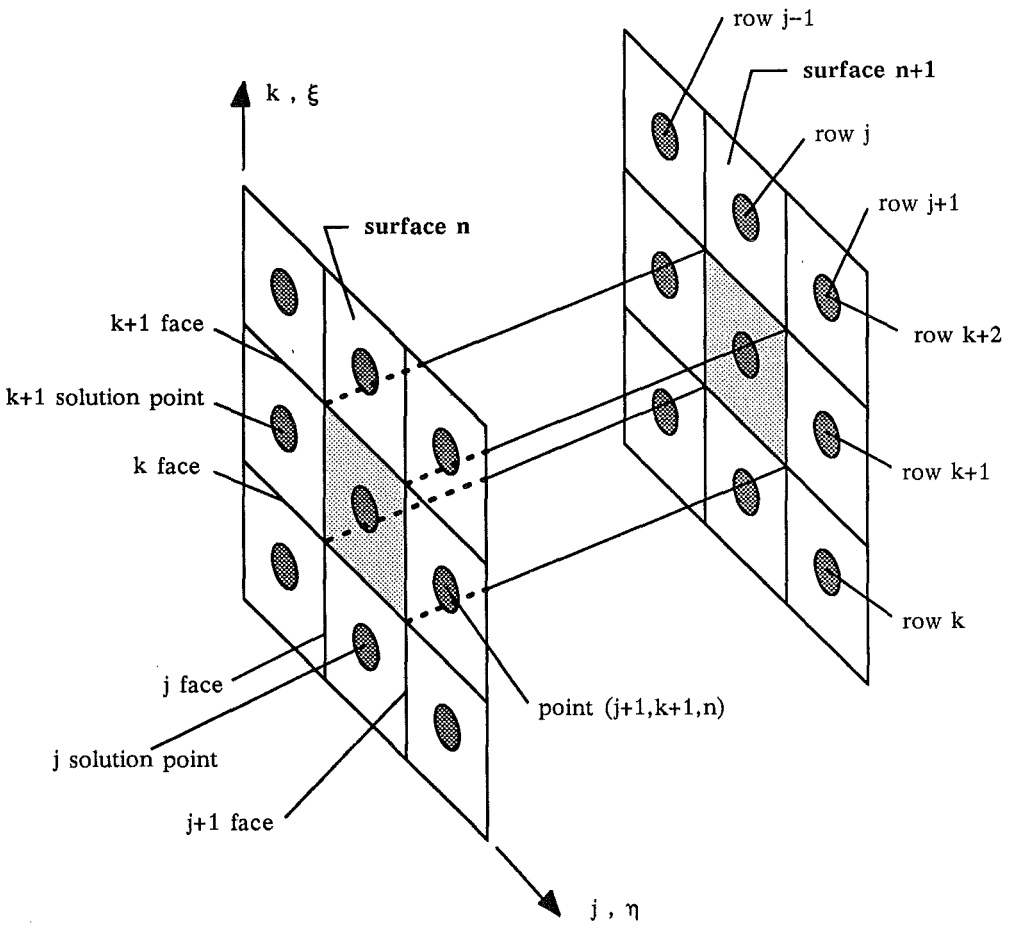


Figure 9. Cell layout for the ALPHA–scheme

From Eq. (3.2), we write

predictor :

$$\tilde{E}^{\overline{n+1}}_{j,k+1} = \tilde{E}^{n}_{j,k+1} - \sum_{i=1}^{5}\left[(\tilde{F}_i)^{n}_{j,k+1} - (\tilde{F}_i)^{n}_{j,k}\right] - \sum_{i=1}^{5}\left[(\tilde{G}_i)^{n}_{j+1,k+1} - (\tilde{G}_i)^{n}_{j,k+1}\right] \tag{3.11}$$

where, for example, on the k^{th} ξ = constant face,

$$(\tilde{F}_i)^{n}_{j,k} = sgn|(\hat{\lambda}_i)^{n}_{j,k}|(\tilde{F}_i)^{n}_{j,k+\sigma_i}$$

with $$(\hat{\lambda}_i)^{n}_{j,k} = \frac{1}{2}[(\lambda_i)^{n}_{j,k+1} + (\lambda_i)^{n}_{j,k}]$$

and $$\sigma_i = \frac{1}{2}\left(1 - sgn(\hat{\lambda}_i)^{n}_{j,k}\right)$$

and $$(\tilde{F}_i)^{n}_{j,k+\sigma_i} = (\hat{\lambda}_i)^{n}_{j,k}(r_i\, l_i^T\, \tilde{E})^{n}_{j,k+\sigma_i}$$

It is understood that quantities such as $(\lambda_i)^{n}_{j,k}$ are evaluated using the primitive variables at solution point (j,k,n) and values of $\bar{\xi}$, $\bar{\zeta}$ on the face in question (eg., in this case $\bar{\xi}^{n}_{k}$, $\bar{\zeta}^{n}_{j}$). This is because $\bar{\xi}$, $\bar{\zeta}$ are face quantities and have no meaning at solution points.

Thus, application of Eq. (3.11) gives $\tilde{E}^{\overline{n+1}}$ as a predicted solution. The decoding routine discussed in section (2.3) will provide the primitive variables at this solution point provided a value for $\bar{\zeta}^{\overline{n+1}}$ is known. This requirement essentially translates to knowing the grid at $\overline{n+1}$ which is obtained by the grid generation scheme once the body and shock boundary points are known. The body points are known at $\overline{n+1}$ by virtue of the QUICK geometry modeling system, and the shock points are known at $\overline{n+1}$ by integrating the n^{th} level shock slope vector, $\vec{r}^{\,n}_{\xi}$, obtained in the shock fitting routine. Thus, decoding is done and all information at $\overline{n+1}$ is available. The corrector step is then taken using

corrector :

$$\tilde{E}^{n+1}_{j,k+1} = \tilde{E}^{n}_{j,k+1} - \sum_{i=1}^{5}\left[(\tilde{F}_i)^{n+1/2}_{j,k+1} - (\tilde{F}_i)^{n+1/2}_{j,k}\right] - \sum_{i=1}^{5}\left[(\tilde{G}_i)^{n+1/2}_{j+1,k+1} - (\tilde{G}_i)^{n+1/2}_{j,k+1}\right]$$

where, for example

$$(\tilde{F}_i)^{n+1/2}_{j,k} = \alpha\, \tilde{F}_{M_i} + (1-\alpha)\, \tilde{F}_{U_i}$$

with $\tilde{F}_{M_i}$ given by Eq. (3.7) and $\tilde{F}_{U_i}$ given by Eqs. (3.9) and (3.10). Note that the direction decision depends on σ_i . This must be based on n^{th} level $\hat{\lambda}_i$ values both in the predictor and the corrector. However, in the evaluation of $(\tilde{F}_i)^{\overline{n+1}}$, such as is needed in Eq. (3.7) the $\hat{\lambda}_i$ used there is obtained from $\overline{n+1}$ data.

3.3 Boundary Conditions

The integration algorithm described in the previous section is applicable to general solution points. There are four boundaries which require special treatment. The two symmetry boundaries are handled with a simple reflection procedure. The body boundary condition influences the solution at the solution point just off the body since computation of this point requires the body flux information. This flux information simply enforces the surface tangency requirement. The solution at the point directly on the body is determined with a half cell matching procedure. Following each predictor and corrector step, the surface tangency condition is enforced explicitly with Abbett's method (Ref. 11).

At the shock surface, the point at k = kx just a half cell inside the shock requires the flux at the shock. The algorithm upwinds normally based on the eigenvalues except that when data is needed outside the shock cell face for the flux at this face, the values on the shock are used instead. The shock point is computed with a half cell matching procedure identical to that at the body. This provides a complete solution at the shock from which only the pressure is kept. The shock fitting routine is then used to enforce the Rankine–Hugoniot jump conditions.

4.0 THE STARTUP PROBLEM

EULER3DS solves the inviscid steady flow equations. Consequently, the initial solution surface from which it starts marching must theoretically be a correct steady state solution. This can be achieved in one of two ways generally. If the nosetip of the body in question is a conical nosetip, then an "arbitrary" initial solution may be marched with **EULER3DS** down this hypothetical conical body (continuation of conical nosetip) until a steady conical solution is reached. This conical solution may then be scaled back to the nosetip station and used as the initial solution surface for the actual body. The second way applies to bodies with non–conical nosetips. In this case a separate computer code is required to solve for the steady state solution as the time asymptotic solution to the unsteady equations. The first of these two ways is the only method utilized in this work.

4.1 Initial Solution

Before **EULER3DS** can begin computing, an initial solution msut be provided. **EULER3DS** reads this initial solution from FORTRAN unit 'IRSTRT'. The sequence of read statements used is

```
READ(IRSTRT,END=4444)JX,KX
READ(IRSTRT)XREFM,YREFM,ZREFM,XFOR,YFOR,ZFOR,XMOM,YMOM,ZMOM
NTOT=JX*KX
NCTOT=(JX-1)*(KX+1)
  READ(IRSTRT)(P(L),RHO(L),U(L),V(L),W(L),L=1,NCTOT)
      ,(X(L),Y(L),L=1,NTOT),Z
```

The information required by these read statements must have been generated and written out in this manner by the user prior to execution of **EULER3DS**.

The first read statement expects the values of JX, KX. These represent the number of points in the 'corner grid' around the body and between body and shock respectively.

The second read statement reads information concerning force and moment loads. The vector (XREFM, YREFM, ZREFM) represents the coordinates of the point about which the moment due to pressure is computed. The vector (XFOR, YFOR, ZFOR) represents the accumulated force vector for that portion of the body upstream of the current station. This vector is structured as (–NORMAL, SIDE, AXIAL). The vector (XMOM, YMOM, ZMOM) is the accumulated moment

about (XREFM, YREFM, ZREFM) due to the pressure forces. Note that the units on these quantities depend upon the units of the code's (x,y,z,p) values. The code's p values are really p/PINF and it's x,y,z units are arbitrary. This results in a force per unit reference area per unit pressure and a moment per unit reference area squared per unit pressure. These are, in effect, force and moment coefficients. Note that the reference area is the square of the physical length of a unit x,y, or z code element.

The third read statement reads the values of pressure, density, and velocity components at each point in the 'cell center grid' (NCTOT points) and the grid coordinates of the 'corner grid' (NTOT points) plus the value of z at this station. See Figure 6 in section (3.1.1) for a reference on the mesh layout.

The user is responsible for providing this information initially to **EULER3DS** to begin computation. **EULER3DS** produces as output on FORTRAN unit 'IRSTRT + 1' subsequent solutions in this format for future startups.

5.0 INPUT DATA

In addition to the initial solution data file on FORTRAN unit 'IRSTRT', **EULER3DS** also reads input from unit 5 and unit 15. The data on unit 15 is geometry definition data which is created by the QUICK geometry modeling system described in Refs. (7–9). A description of this package is beyond the scope of this report. FORTRAN unit 5 is read by **EULER3DS** to provide various parameters which control various aspects of the code. These parameters are described in the next subsection.

5.1 Input Parameter Description

The input parameters required by **EULER3DS** are cast into two basic categories. These are 'general input' and 'grid related input'. The 'general input' supplies parameters via the FORTRAN NAMELIST called 'NAMLST' while 'grid related input' is supplied through a FORTRAN NAMELIST called 'GRDNAM'. The parameters contained in 'NAMLST' are now listed with their defaults indicated in parantheses and a brief description of each follows.

IRSTRT	(10)
NGAS	(0)
IORDER	(1)
ISHOCK	(3)
NSTEP	(1)
NOUT	(1)
IOUT	(0)
NGRD	(100)
ALFA	(0.5)
CN	(0.95)
FSMACH	(no default)
PINF	(1.0)
RHOINF	(1.0)
RELAX	(1.0)
GAM	(1.4)
ALPHA	(0.)
NPLT	(10000)
NDSK	(10000)

IRSTRT: FORTRAN unit number for initial data surface (NOTE : output solution is on unit irstrt+1)
NGAS: (= 0 for perfect gas) or (= 1 for equilibrium air real gas)
IORDER: (= 1 for first order) or (= 2 for second order)
ISHOCK: (= 1 for shock point propagation along existing η = constant lines)
or (= 2 for propagation along projection of shock normal onto ζ = constant surface)
or (= 3 for propagation along cylindrical rays)
NSTEP: number of marching steps this run
NOUT: print out frequency on unit 6 (print every NOUT steps)

IOUT: [0=abbreviated print out (body–shock data only)] [1=full flowfield printout]
NGRD: max number of iterations in grid construction
ALFA: alpha scheme parameter (0.= pure upwind, 1.= modified Mac.) not used when iorder=1
CN: Courant number. max value=1 for stability
FSMACH: freestream Mach no.(no default)
PINFSI: freestream pressure in SI units(needed for real gas option only! do not include for perfect gas)
RINFSI: freestream density in SI units(needed for real gas option only! do not include for perfect gas)
RELAX: relaxation parameter used in grid line relaxation scheme
GAM: perfect gas specific heat ratio
ALPHA: angle of attack in degrees
NPLT: plot output frequency for genplot (plot every NPLT steps)
NDSK: restart solution output frequency (write out every NDSK steps)

Next, the parameters contained in 'GRDNAM' are listed with their defaults indicated in parentheses and a brief description of each follows.

MGRD	(4)
EPSSP	(no default)
EPSCP	(no default)

MGRD: (1=elliptical Laplace's equation grid)
or (2=elliptical Poisson equation (p/q by Ref 3.))
or (3=algebraic grid with Laplace smoothing (Ref 5.))
or (4=simple algebraic grid(equal space from body to shock))
or (5=elliptical Poisson equation (p/q by Ref 6.))
EPSSP: Used only when MGRD=3
(0. = no orthogonality at wall)
or (1. = max orthogonality at wall)
EPSCP: Used only when MGRD=3
(0. = no clustering toward wall)
or (1. = max clustering toward wall)

NOTE: For option 5, the p/q control functions are based on a requirement of orthogonality at the body and first cell spacing of the initial grid estimate before relaxation. This may not be satisfactory in all cases. The requirement of orthogonality is met by bisecting the angle between (j–1,j) and (j,j+1) segments. This procedure gives reasonable results even at a pointed tip.

The only parameters one should usually need to prescribe to run a perfect gas solution are:

IORDER (if second order is desired)
NSTEP
NOUT
FSMACH
ALPHA

6.0 SUMMARY

EULER3DS is a robust 3–D space marching code for external inviscid supersonic flows about arbitrary geometries. The code has a variety of options for various circumstances. The use of IORDER=1 is recommended for situations with exceptionally strong shocks, or steep gradients, particularly during sudden startups. Once this sudden behavior has passed, the user may switch to IORDER=2.

Caution must be used in choosing the parameter ISHOCK. One helpful feature of **EULER3DS** is that points on the body contour are spaced based on equal arclength. This maintains a uniformity in the body grid. The shock point spacing, however, is an ever evolving situation depending on the shock shape and its evolution in the ζ direction, and on the way the shock points are propagated. The method with the greatest chance of success for an arbitrary problem is ISHOCK=3 (i.e., propagation along cylindrical rays in a ζ = constant plane. When this occasionally causes shock points to be-

come spaced improperly over some ζ distance, the user must explore the other options. The author has developed a method which looks very promising for propagating shock points based on the shape of the body contour. This option is not present in **EULER3DS** however.

Regarding the grid generation schemes, the best advice is to use the simplest scheme which will work for the particular problem being solved. The order of preference would be MGRD=4,3,1,2,5. MGRD=4,3,or 1 will work for most problems. MGRD=2 and 5 represent a level of sophistication which results in clustering capability at the expense of robustness for some shapes.

7.0 REFERENCES

1. Srinivasan, S., Tannehill, J.C., and Weilmuenster, K.J., "Simplified Curve Fits for the Thermodynamic Properties of Equilibrium Air," NASA RP-1181, August 1987.

2. Winslow, A., "Numerical Solution of the Quasilinear Poisson Equation," Journal of Computational Physics, Vol. 1., pp 149-172, 1966.

3. Middlecoff, J.F., and Thomas, P.D., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," AIAA Paper No. 79-1462, Williamsburg, Virginia, 1979.

4. Thompson, J.F., Thames, F.C., and Mastin, C.M., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, Vol. 15., pp 299-319, 1974.

5. Noack, R.W., "Inviscid Flow Field Analysis of Maneuvering Reentry Vehicles Using the SCM Formulation and Parabolic Grid Generation," AIAA Paper No. 85-1682, AIAA 18th Fluid dynamics and Plasmadynamics and Lasers Conference, Cincinnati, Ohio, July, 1985.

6. Sorenson, R.L., and Steger, J.L., "Grid Generation in Three Dimensions by Poisson Equations with Control of Cell Size and Skewness at Boundary Surfaces," Advances in Grid Generation, FED-Vol. 5, Ed. K.N. Ghia and U. Ghia, ASME Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, 1983.

7. Vachris, A., and Yaeger, L., "QUICK-GEOMETRY User's Manual." Grumman Aerospace/ Aerodynamic Section Technical Data Report No. 393-74-1, 1974.

8. Vachris, Alfres F., and Yaeger, Larry S., "QUICK-GEOMETRY: A Rapid Response Method for Mathematically Modeling Configuration Geometry," NASA SP-390, pp 49-73, 1975.

9. Shope, Frederick L., "Simplified Input for Certain Aerodynamic Configurations to the Grumman QUICK-GEOMETRY System (A Prekwik User's Manual)," AEDC-TR-77-62, August 1977.

10. Fromm, J.E., "A Method for Reducing Dispersion in Convective Difference Schemes," Journal of Computational Physics, Vol. 3., pp 176-189, 1968.

11. Abbett, M.J., "Boundary Condition Computational Procedures for Inviscid Supersonic Steady Flow Field Calculations," Aerotherm Corporation Final Report 71-41, Mountain View, California, 1971.